第九期 | OmniGraffle 复刻你的键盘 - 少数派

第九期 | OmniGraffle 复刻你的键盘

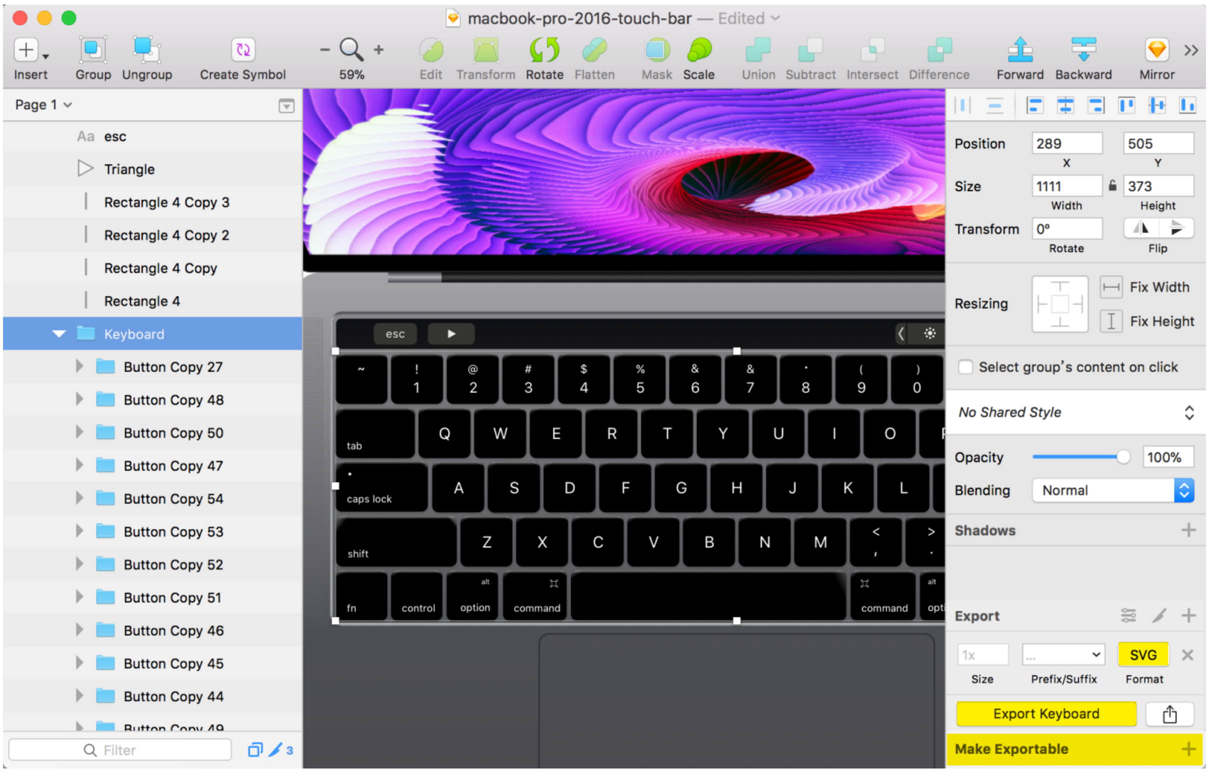
在 OmniGraffle 里复刻一份键盘对初学者来说有点像是一个测验，因为这个过程中会涉及到资源的搜集、导入、群组、对齐、字体、近似对象的选择、快速定位、绘制简单图形、型版等等功能点和操作，加上工作量也摆在那里，既考验耐心也检验我们对工具掌握的纯熟程度。操作效率和技巧很多时候是经历痛苦「折磨」出来的。

键盘图里包含很多重复的对象，涉及到布局和可参照的实际目标，可以说是一个很合适的 OmniGraffle 绘图打磨目标，低头看看自己的键盘就知道还需要修正那些地方。然而从 0 开始还是会让人觉的不着边际，所以我们先找一个键盘。

导入 SVG 资源

学习双拼输入法的时候在网上找过不少键位图，大部分是位图格式的，分辨率也不高，放在屏幕上看看还行，如果要打印出来会很模糊。原来的键盘图来源已经不可考，为了写教程重新在 Sketch 资源网找到了一份，我们就基于这个 MacBook Pro 2016 Sketch Resource Sketch 文件开始，没有安装 Sketch 应用的也不用担心，文章最后的附件里也单独提供了一份导出的 SVG 键盘图。

Sketch 文件导出为 SVG 并不能完美的保留所有的绘图要素属性，字体、渐变、阴影等等都会发生变化，越是复杂的群组对象这方面的问题越突出，当我们用 OmniGraffle 打开时文本可能都挤在一起。建议用 Sketch 导出时只选择必要的部分，降低构成对象的复杂度，另外，根据导入后字体的兼容情况，有选择性的将部分文本「Convert to Outlines」转曲线，这个过程是最繁琐的，没有快捷方式，转完后还需要查验一下转换的效果。


macbook-pro-2016-touch-bar — Edited
Insert
Group
Ungroup
Create Symbol
59%
Edit
Transform
Rotate
Flatten
Mask
Scale
Union
Subtract
Intersect
Difference
Forward
Backward
Mirror
Page 1
esc
Triangle
Rectangle 4 Copy 3
Rectangle 4 Copy 2
Rectangle 4 Copy
Rectangle 4
Keyboard
Button Copy 27
Button Copy 48
Button Copy 50
Button Copy 47
Button Copy 54
Button Copy 53
Button Copy 52
Button Copy 51
Button Copy 46
Button Copy 45
Button Copy 44
Filter
Resizing
Fix Width
Fix Height
Shadows
Export
Size
Prefix/Suffix
Format
Export Keyboard
Make Exportable

Position 289 X 505 Y

Size 1111 Width 373 Height

Transform 0° Rotate Flip

☐ Select group's content on click

No Shared Style

Opacity 100%

Blending Normal

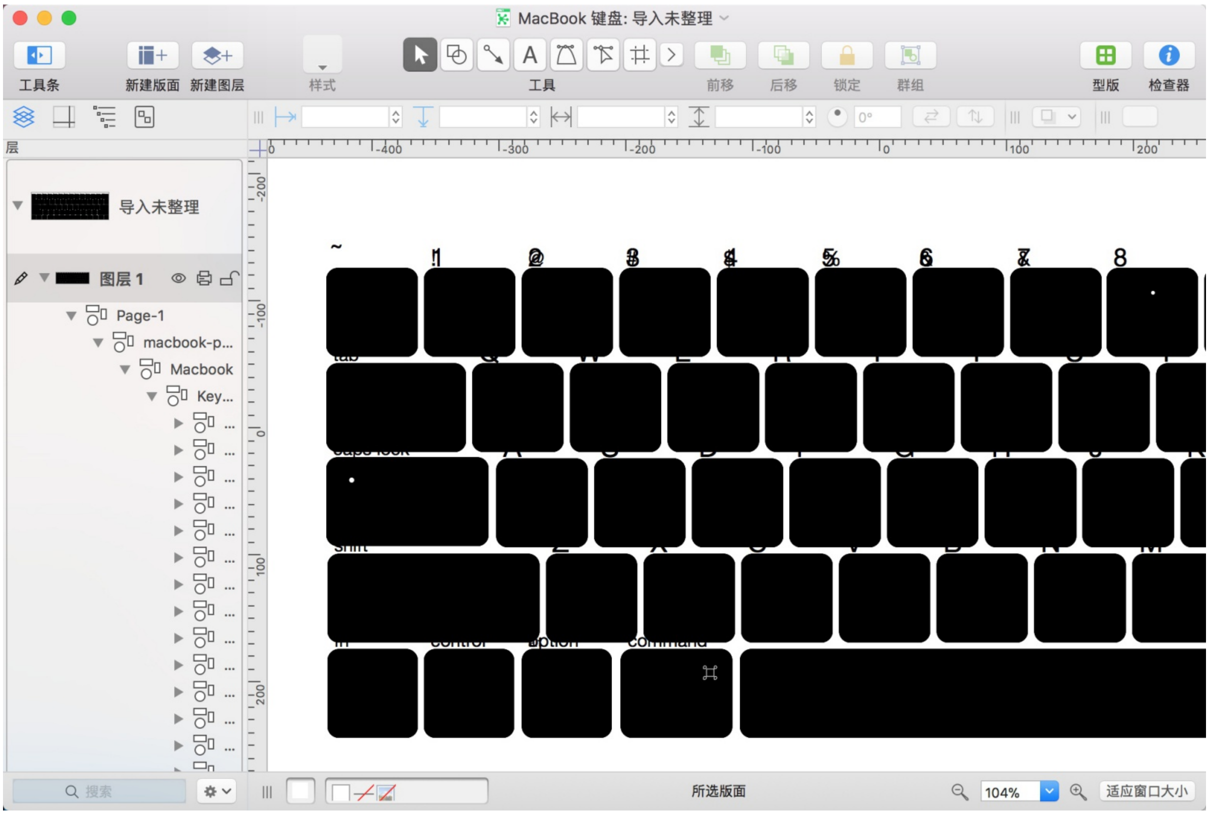
Export 1x SVG

整理好 Sketch 中需要导出的内容后，选择它创建切片并导出，例如，只选择文件中的 Keyboard 部分导出，操作步骤如下：

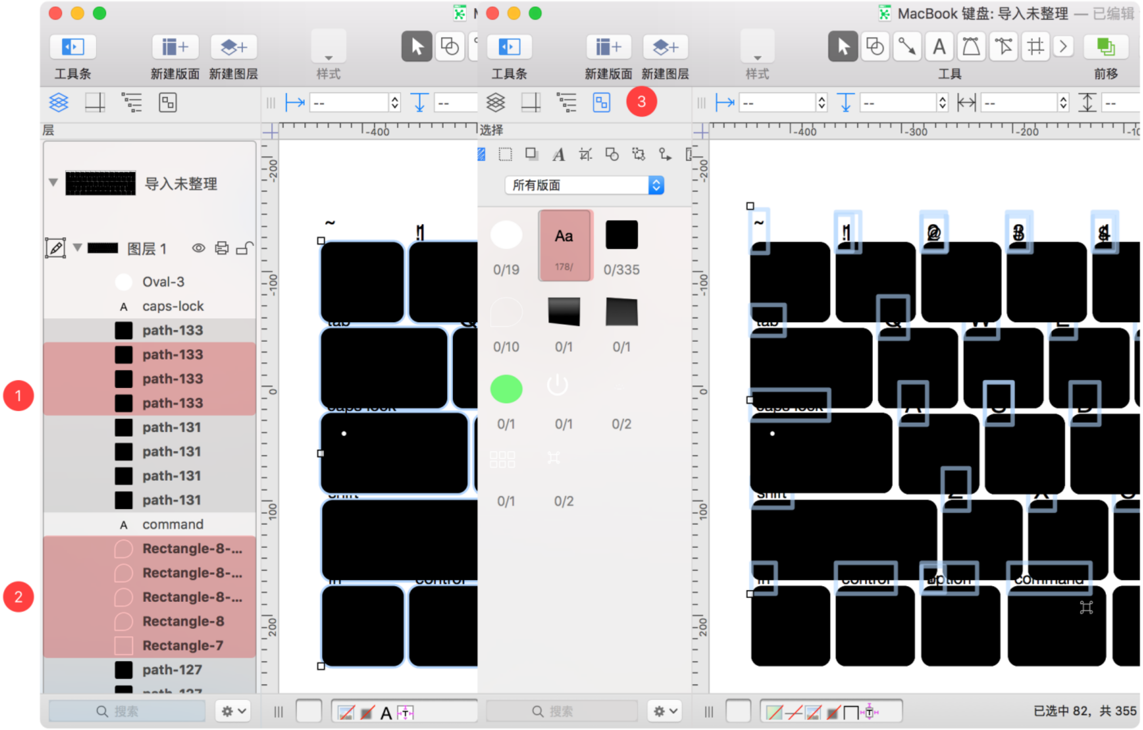
1. 左侧 Page 列表中选择要导出的部分，例如 Keyboard
2. 右侧底栏点击「Make Exportable」旁的 + 号
3. Export 中选择导出格式为 SVG，点击导出按钮「Export Keyboard」

打开 OmniGraffle 新建空白文稿，由菜单选择「文件 - 放置图像」导入上一步准备好的 keyboard.svg 文件。

导入的键盘图看上去黑乎乎一片，这是因为字体颜色和形状填充都被 OmniGraffle 重置了，从左侧的层列表中可以观察到，整个 SVG 是以单一群组的方式导入的，而且还嵌套了很多群组对象。键盘按键上的字符虽然乱掉了，不过布局没问题，算是一个不错的开始。


MacBook 键盘: 导入未整理
工具条
新建版面
新建图层
样式
工具
前移
后移
锁定
群组
型版
检查器
层
导入未整理
图层 1
Page-1
macbook-p...
Macbook
Key...
搜索
所选版面
104%
适应窗口大小

面对这样的导入情形，后续还有不少工作需要完成。字体颜色需要调整，键盘字符摆放、对齐，所有这些操作都是面向单个对象的，所以整理的第一步是「解散群组」，反复的全选操作解散群组，直到所有的对象都被扁平化到同一个层级。


工具条
新建版面
新建图层
样式
MacBook 键盘: 导入未整理 — 已编辑
工具
前移
层
选择
导入未整理
图层 1
Oval-3
caps-lock
path-133
path-133
path-133
path-133
path-131
path-131
path-131
path-131
command
Rectangle-8-...
Rectangle-8-...
Rectangle-8-...
Rectangle-8
Rectangle-7
path-127
所有版面
0/19
0/335
0/10
0/1
0/1
0/1
0/1
0/2
0/1
0/2
1
2
3
搜索
已选中 82，共 355

所有的群组对象被打散成基础对象后，就成了上图所示的效果。为什么非要都解散群组呢？主要是为了选择单个的对象方便，全部解散后在「近似对象选择」面板 ③ 中就能快速的选择同一种属性的元素，例如，选择所有文本对象，同理如果我们要改变黑色键盘的颜色在这里选择也很方便。

观察解散群组后的对象还发现了几个问题：

- ⌘ 符号原图里是一个群组对象，解散群组 ② 后破坏了它们的整体性，为避免其他操作最后把 cmd 符号搞丢了，优先选择将这几个构成形状还原成群组对象；
- 每个键位 ① 是 4 个叠在一起的形状组成的，对照 Sketch 原始素材发现，设计图的作者给键位增加光泽和立体效果，左右角上的光效，立体感的渐变用到了 3 个尺寸一样的形状进行叠加，转成 SVG 格式后这些效果都被重置成了单色填充。

键位形状的重复并不影响输出效果，这个坑大家可以选择略过，范例里已经单独保存了一份去掉重复形状的键盘图。

包含很多对象的绘图场景中，会频繁用到选择、锁定、群组、定位、对齐以及对象顺序调整。

对象的批量操作

选择对象最直观的方式是将画布缩放到合适的比例，直接用鼠标指针圈选对象，以画布空白

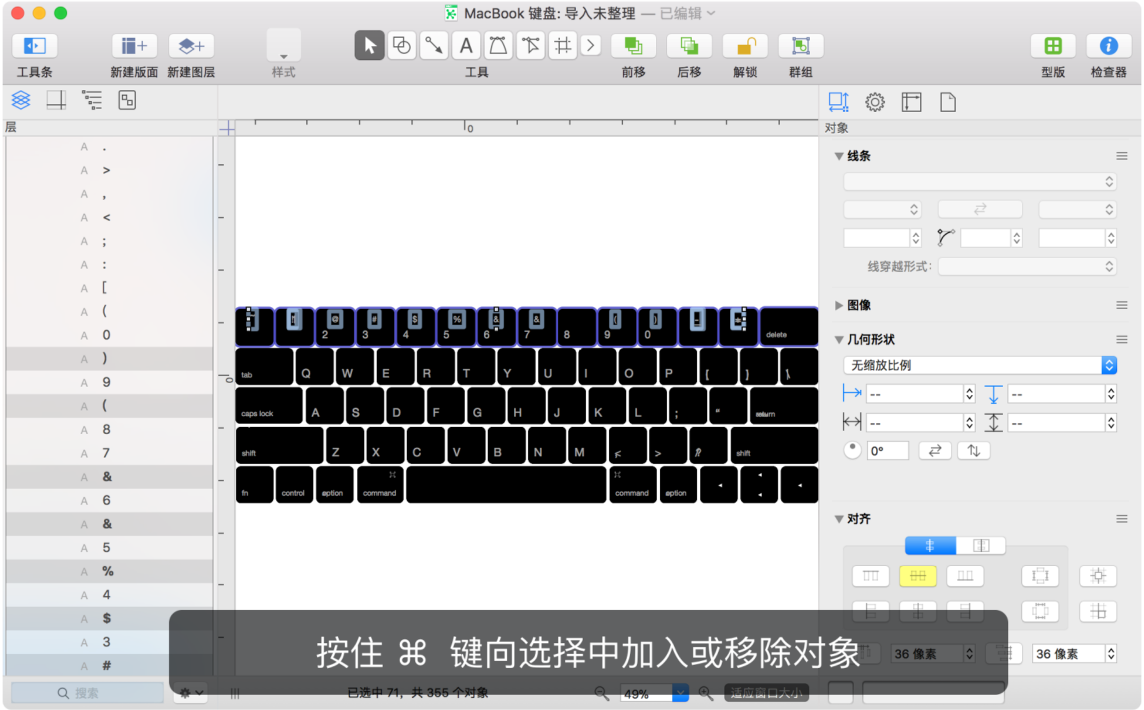
的地方作为起点，鼠标指针划过的对象都会被选中。如果想单独选择前景中的对象，可以按住 Option 再进行圈选，这样只有完全被圈中的对象才会被选中。

要调整键盘图里错位的键盘字符，需要多次选择多个对象，会用到不同的选择技巧。选择键位上的字符时可以采用按住 Option 圈选的方式，但是这样的选择效率不太高，依旧有可能选择到下方的键位。如何避免这种问题呢？用前面提到的近似对象选择选择所有的键位形状，然后右键「锁定」。

- 锁定键位后，可以用 ^A 全选所有对象来移动键盘上的字符，因为键位已经锁定所以移动的就都是键位以外的其他对象；
- 为便于识别，先给所有字符换个颜色，并将其统一移到最前（⇧⌘F），没有了键位的遮挡，下一步操作才方便；
- 全部字符位置初调后，还有一些是重叠在一起的，例如，第一排的标点符号和数字。这个时候需要用到层列表模式里的选择。

横向圈选第一排的字符，按住 ⌘ 键从层视图的对象列表中去掉这次不想选中的对象，例如，先移除数字的选择，只保留符号。比起在画布中的圈选方式，层、大纲、近似对象几种模式更适合精确选择对象，和圈选方式结合起来使用对批量对象的操作十分有帮助。

圈选时先选中（或划过）的对象是后面对象的对齐基准


MacBook 键盘: 导入未整理 — 已编辑
工具条
新建版面
新建图层
样式
工具
前移
后移
解锁
群组
型版
检查器
层
对象
线条
线穿越形式：
图像
几何形状
无缩放比例
对齐
搜索
49%
适应窗口大小

按住 ⌘ 键向选择中加入或移除对象

已选中 71，共 355 个对象

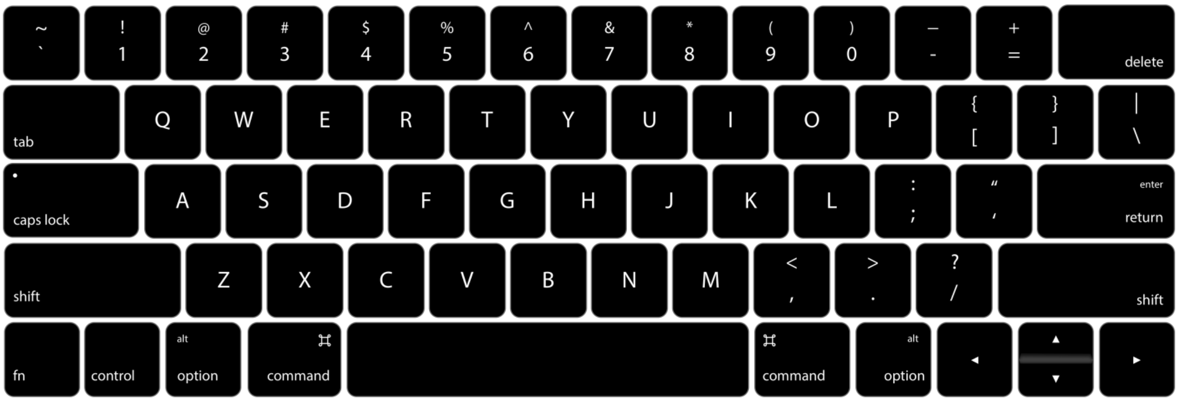
36 像素

36 像素

如何快速的对齐这些标点符号呢？这里可以把底下的键位作为基准来实现：

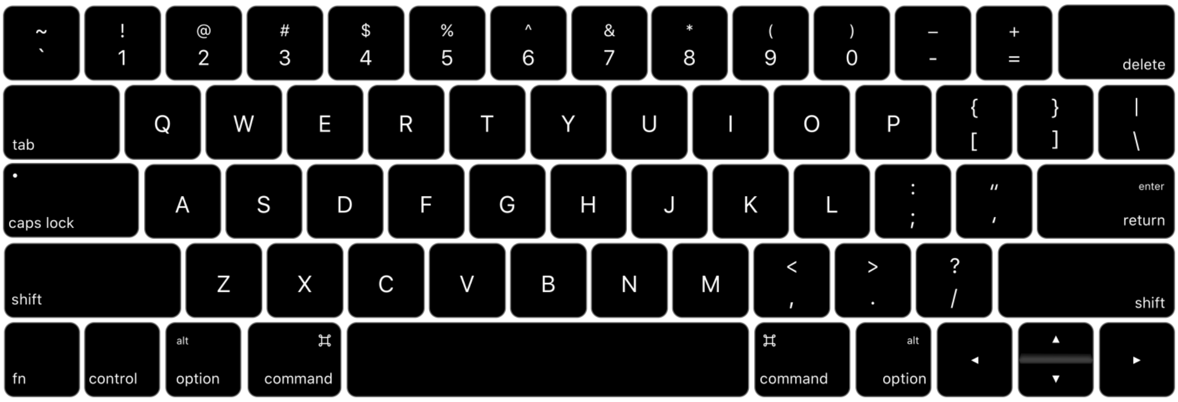
- 用近似对象选择的方式，选择所有键位并解锁（锁定对象没法参与对齐）；
- 先对齐两头的目标，分别单独对齐 ~ 和 + 号和它所在的键位形状，对齐：垂直居中；
- 左侧的层模式列表中选择第一排的标点符号（~ 到 +），对齐：水平居中，对齐：平均水平间距。~ 符号已经单独对齐过，选择它作为后面对象的水平居中基准，+ 也单独对齐过，+ 号和 ~ 号共同约束它们之间对象的平均分布位置；
- 其他几排的字符采用同样的方式操作。

整理好之后的样子，是不是很有成就感！如果都已经对齐了，看上去还不对，就只剩下字体的问题了，Apple 用在键盘上的显示字体是「Myriad Pro」。


~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 – - + = delete
tab Q W E R T Y U I O P { [}] | \
caps lock A S D F G H J K L : ; " ' enter return
shift Z X C V B N M < , > . ? / shift
fn control alt option ⌘ command ⌘ command alt option ◂ ▴ ▾ ▸

字体：Myriad Pro

换成 SF UI Text 字体看看效果也不错。


~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 – - + = delete
tab Q W E R T Y U I O P { [}] | \
caps lock A S D F G H J K L : ; " ' enter return
shift Z X C V B N M < , > . ? / shift
fn control alt option ⌘ command ⌘ command alt option ◂ ▴ ▾ ▸

字体：SF UI Text

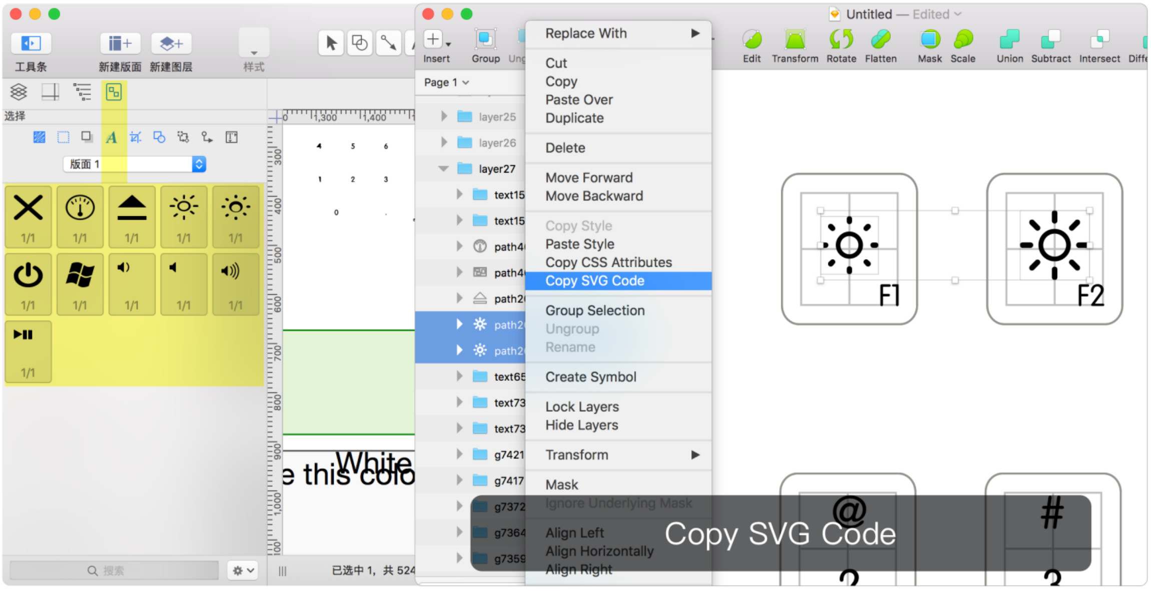
> 锁定对象不会响应选择或者对齐这些操作，为了避免误移动对象，在对齐的过程中可能需要分组多次的进行锁定（⌘L）和解锁（⌥⌘L），频繁的操作记住快捷键效率会高一些。

补充设计图没有的功能区

网上下载的Sketch 设计图 是 Touch Bar 版本的 MacBook Pro，没有传统键盘最上面的一排功能键。缺失的形状并不复杂，所以接下来需要在 OmniGraffle 绘制这些形状，让键盘图变得更完整。

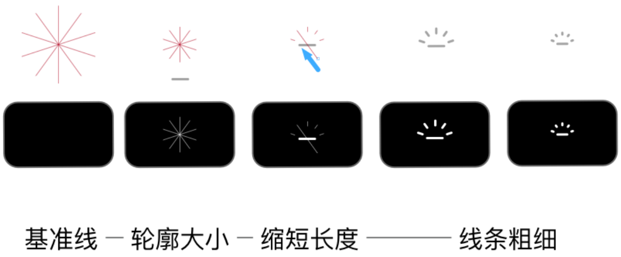
同样，先在网上找找可以用的资源，比如这个 [Apple Keyboard Layout for WASD V2 Keyboards]() 、以及这个 [www.flaticon.com/free-icons]()，我们只需要找到几个功能符号图标就行。

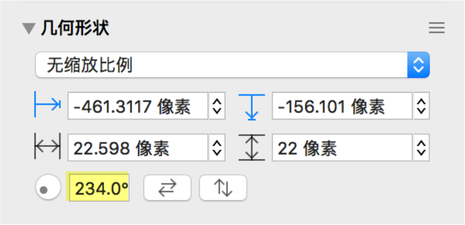
用 OmniGraffle 打开这个 Apple Keyboard Layout for WASD V2 Keyboards.svg， 不要被里面的混乱吓到，解散群组后依旧是在「近似对象选择」里找我们想要的符号标志，复制出来就好。当然有条件的话用 Sketch 打开内容显示会更完美，直接选择需要的形状「Copy SVG Code」，然后到 OmniGraffle 里粘贴，简单对象都可以采用这种导入的方式。


工具条
新建版面
新建图层
样式
选择
版面 1
Insert
Group
Page 1
layer25
layer26
layer27
Replace With
Cut
Copy
Paste Over
Duplicate
Delete
Move Forward
Move Backward
Copy Style
Paste Style
Copy CSS Attributes
Copy SVG Code
Group Selection
Ungroup
Rename
Create Symbol
Lock Layers
Hide Layers
Transform
Mask
Ignore Underlying Mask
Align Left
Align Horizontally
Align Right
Untitled — Edited
Edit
Transform
Rotate
Flatten
Mask
Scale
Union
Subtract
Intersect
F1
F2
@
#
Copy SVG Code
已选中 1，共 524

网上找到的素材里缺一个键盘亮度调节的图标，考虑在 OmniGraffle 里绘制。

- 首先，复制一个键位做大小的基准；
- 用形状中的「线条」做一个基准线，亮度图标里围绕的一圈是 10 等分的角度，用 5 根线条（形状线条的文本位置设定推荐选择适合文本和环绕形状）来实现，绘制一根后复制 5 个对齐和居中让它们完全重叠在一起；通过层模式选择每一根，按照 36° 的增量在「几何形状」里设置角度，90、126、162、198、234。基准线是绘图里很常用的一种手段，有了基准后面就方便对齐和布局了；


基准线 — 轮廓大小 — 缩短长度 ——— 线条粗细



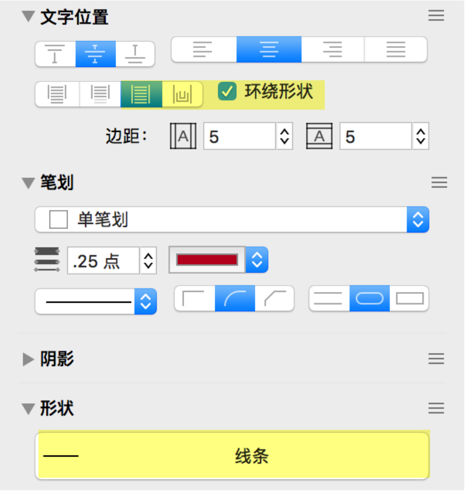
几何形状

无缩放比例

-461.3117 像素 -156.101 像素

22.598 像素 22 像素

234.0°



文字位置

☑ 环绕形状

边距： 5 5

笔划

☐ 单笔划

.25 点

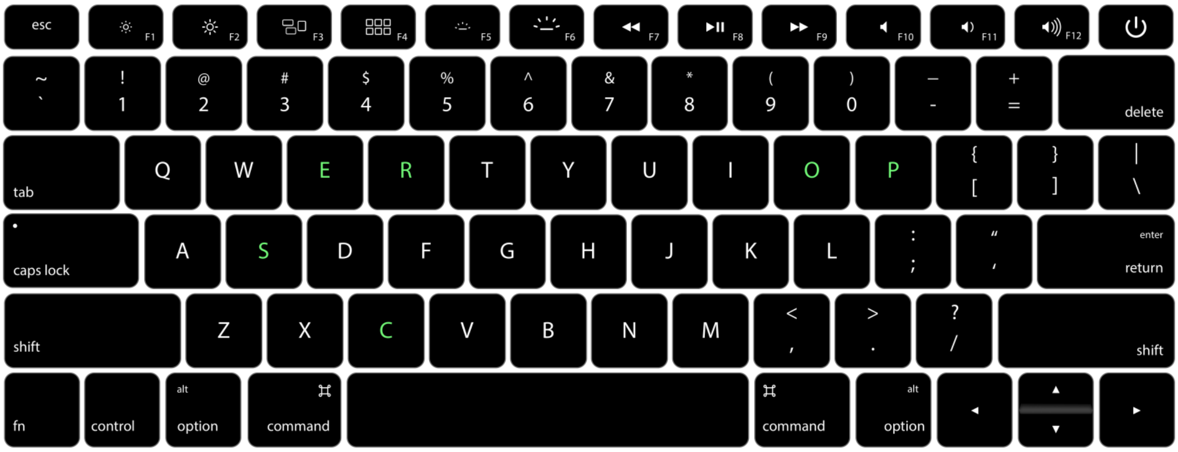
阴影

形状

线条

- 群组 5 根线条，基于键位的大小缩小到合适尺寸；
- 解散群组，缩短每根线条，从下往上，留下需要的长度，例如 3 px，拖动缩小到近似值即可，精确数值可在几何形状中设置；
- 最后群组，调整线条笔划的粗细，例如 .25 ，改个颜色摆放到键位上就完成了。

其他几个键盘符号都不复杂，都可以自己画一画练练手。和前面的键盘组合在一起看看结果。


esc F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 — - + = delete
tab Q W E R T Y U I O P { [}] | \
caps lock A S D F G H J K L : ; " ' enter return
shift Z X C V B N M < , > . ? / shift
fn control alt option ⌘ command ⌘ command alt option

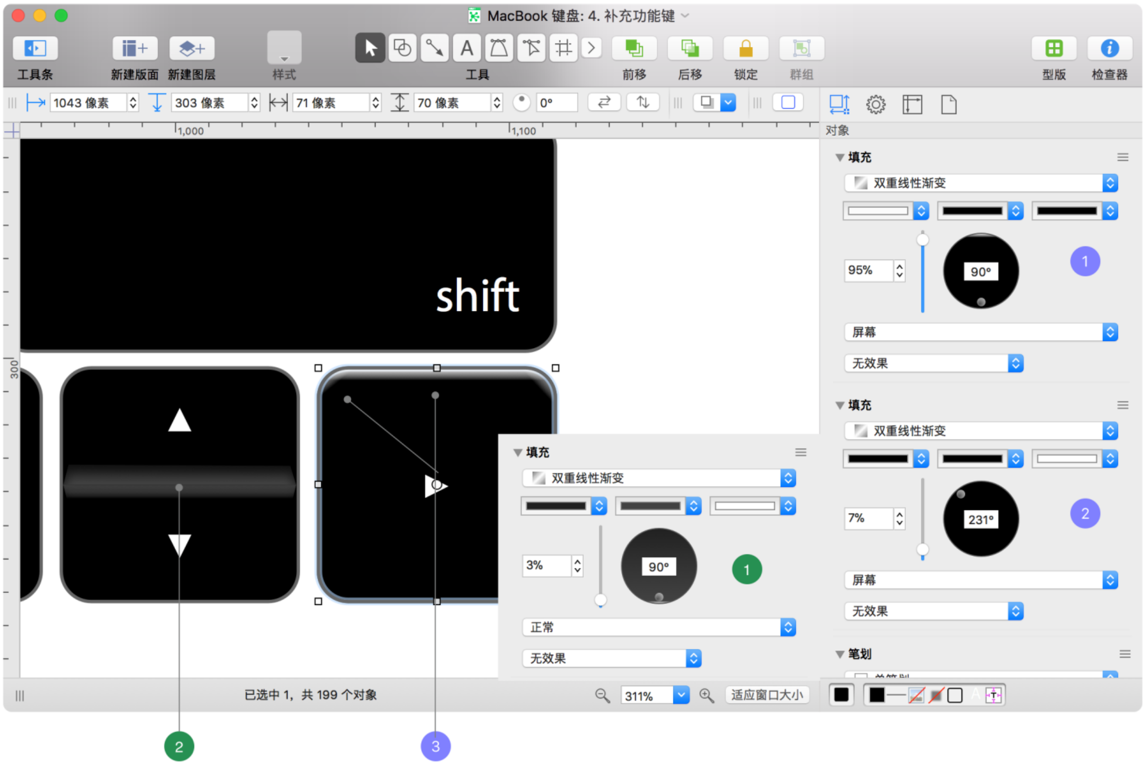
有了这个基础的键盘图，基于它作出变化，制作双拼键位图就没有什么难度了，不妨自己尝试制作一个双拼的键位图，只需要保留字母区域（Qz），注意几个问题：

- 建议采用图层的方式来管理绘制，例如无刻的键位单独一个图层，不同的双拼方案单独一个图层；
- 基于图层的锁定来减少不同类型对象之间的干扰，例如，键位、声母、韵母；

- 通过文本偏移量来控制形状中文本的位置；
- 在系统偏好中定义一个「缩放到所选内容」的快捷键（如：⌥⌘J），这个操作使用频繁。

补充一个「屠龙技」：键盘的立体感

前面处理 Sketch 键盘设计稿的时候曾经遇到过键位形状重复的问题，那些重复的键位形状原本是单独的特效，只是转成 SVG 后 OmniGraffle 无法识别和还原导致的。是否需要给键位加上立体效果要看使用的场合，如下图所示 OmniGraffle 中的立体效果是通过填充来实现的。


MacBook 键盘: 4. 补充功能键
工具条
新建版面
新建图层
样式
工具
前移
后移
锁定
群组
型版
检查器
1043 像素
303 像素
71 像素
70 像素
0°
对象
1,000
1,100
300
shift
填充
双重线性渐变
95%
90°
1
屏幕
无效果
填充
双重线性渐变
7%
231°
2
屏幕
无效果
笔划
填充
双重线性渐变
3%
90°
1
正常
无效果
已选中 1，共 199 个对象
311%
适应窗口大小
2
3

上下箭头中间的部分是两个对称的线性填充，然后用点编辑工具把矩形调成梯形，最终形成视觉上的凹陷感。键位边缘的立体感方式一样，夸张一点还可以给两个角也加上一点光感的效果。学会这个只需要掌握双重线性渐变即可。

附上范例资源：[点此下载（2.9 MB）]

PS：不要找我要字体哦，即便你没用开发者帐号网上也能找到这些字体；）